ICOMP: An Intelligent Layout Compactor

by

Tadashi Matsumoto

Dept. of Information and Computer Science
University of California at Irvine
Irvine, CA 92717

Technical Paper 88-03

ABSTRACT

This paper describes a two phase virtual grid compactor. In the first phase, symbols are arranged according to layout design rules. In the second phase, groups of symbols are compacted. (Symbols with a common signal form a group.) The compactor has achieved an approximate twenty to forty percent gain in test cases.

ICOMP: An Intelligent Layout Compactor

Tadashi Matsumoto

February 11, 1988

1 Introduction

The compactor reads symbolic layout files from LES [LG87], given a particular technology, and sends its output to a CIF file. The compactor can be divided into two phases. (See Fig.1) In the first (rearrangement) phase, symbols are rearranged on a virtual grid according to rules stored in the knowledge base. In the second (compaction) phase, whole groups of symbols are compacted into a CIF file with a fine grid.

2 Symbolic Rearrangement

Unlike other compactors that merely calculate space between symbols—our compactor, by applying expert rules, attempts to reduce the space by rearranging symbols on a virtual grid.

A drawback of rule-based programs is that an increase in the number of working memory instances greatly slows down the execution speed, due to the nature of the search required. However, to optimize the layout locally, we only need data for a few grids around the focus of optimization. This suggests the use of a sliding window. Our window covers only three columns of the layout data at a time.

To apply expert rules, we use a greedy algorithm which slides the window right, grid by grid, until it reaches the end of the layout. Examples of symbol rearrangements are *offsetting, sliding* and *moving* contacts.

We use Figure 2 to explain the application of our heuristic rules. Figure 2a shows a (3 micron process rule) layout before symbol rearrangement. After 10 rules were applied, we get the final mask layout as shown in Figure 2b. We show here three examples of heuristics used in this refinement process. The first one is offsetting contacts. In region (i), we offset two contacts on the same grids to reduce the distance between columns 3 and 4 by 2 microns and the distance between rows 1 and 2 by 1 micron. The second heuristic slides contacts along an existing wire as illustrated in region (ii). In this case, by pushing down contact A and offsetting B, we gain 5.5 microns between columns 5 and 6, and another 1 micron between rows 6 and 7.

Region (iii) shows the third heuristic which moves a contact to an empty space. This rule extends metal and diffusion layers as shown in the figure. This example shows that both the row and column distances are reduced in one pass of the sliding window. In this particular case, we gained 11 microns in the horizontal direction and 3 microns in the vertical direction. This resulted in a twenty percent reduction in layout area.

3 Compaction

Compaction follows symbol rearrangement. Symbols are first organized into groups, then compacted. In the vertical direction, each group consists of all abutting symbols with the same signal. Similarly, in the horizontal direction, except that drains, gates and sources of transistors are treated as if they had the same signals.

The location of a symbol is represented by its virtual grid address. The virtual grid coordinates allow the compactor to access all the symbols adjoining a particular symbol of interest. But the compactor is not restricted to the virtual grid, because it writes directly to the fine grid CIF file.

The object of compaction is the group. The compactor calculates distances between pairs of groups. First the compactor proceeds left to right horizontally, pushing each group as far left as possible. Then it compacts vertically from the bottom to the top.

Two types of transistors may be used during compaction. If the transistor width specified in the symbolic file is smaller than the value defined in the technology file, the

compactor will simply use a lined up transistor. But if the transistor width is greater, the compactor will introduce *S* or *snake* transistors to balance the length loss both horizontally and vertically. An example of snake transistors is shown in Fig.3.

4 Technology File

The technology file consists of two parts. The first part is a definition of symbols under a certain process, which occurs in symbolic files. Another one is minimum space required by the process between any pair of symbols. By changing the contents of the technology file, a different layout is produced under a different process rule.

5 Results of Experiments

The compactor is coded in OPS83 and C. Table 1 shows the statistics of the compactor. Symbol rearrangement resulted in about five percent reduction in area. After group compaction, final layout area was reduced to between sixty and eighty percent of the area of previously developed virtual grid compactors under the typical 2 micron process rule. Figure 4 shows examples of a previous compactor and group compactor with symbol rearrangements. This circuit's statistics are in Table 1 under the name *sun1.t.*

6 Conclusion

We have demonstrated a two phase compactor that is able to improve the silicon efficiency of a virtual grid symbolic layout. The two phases are: symbol rearrangement, and group compaction.

7 Future Research

We believe that knowlege based programming is a winning strategy. To get better results, we will need to integrate circuit designers' expertise into the compaction program. Several interrelated problems that will have to be solved are listed below.

7.1 Transistor Sizing

Transistor shapes should be determined by examination of neighboring symbols at a high level.

7.2 Jog Insertion

A more intelligent jog insertion mechanism is needed. Currently, jogs are introduced to release critical paths, even at the expense of pushing away other symbols. This method often results in a larger final layout.

7.3 Two Dimensional Compaction

Our compaction algorithm only compacts one dimension at a time. In order to achieve smaller area, it will be necessary to consider tradeoffs in two dimensions.

8 Acknowledgements

I would like to thank Steve Lin and Daniel Gajski for many useful discussions during the course of this work. My thanks also go to Jim Fradkin for helping with the production of this report.

References

[LG87] Y. L. Lin and D. D. Gajski. Les: a layout expert system. In *24th Design Automation Conference*, pages 672–678, 1987.

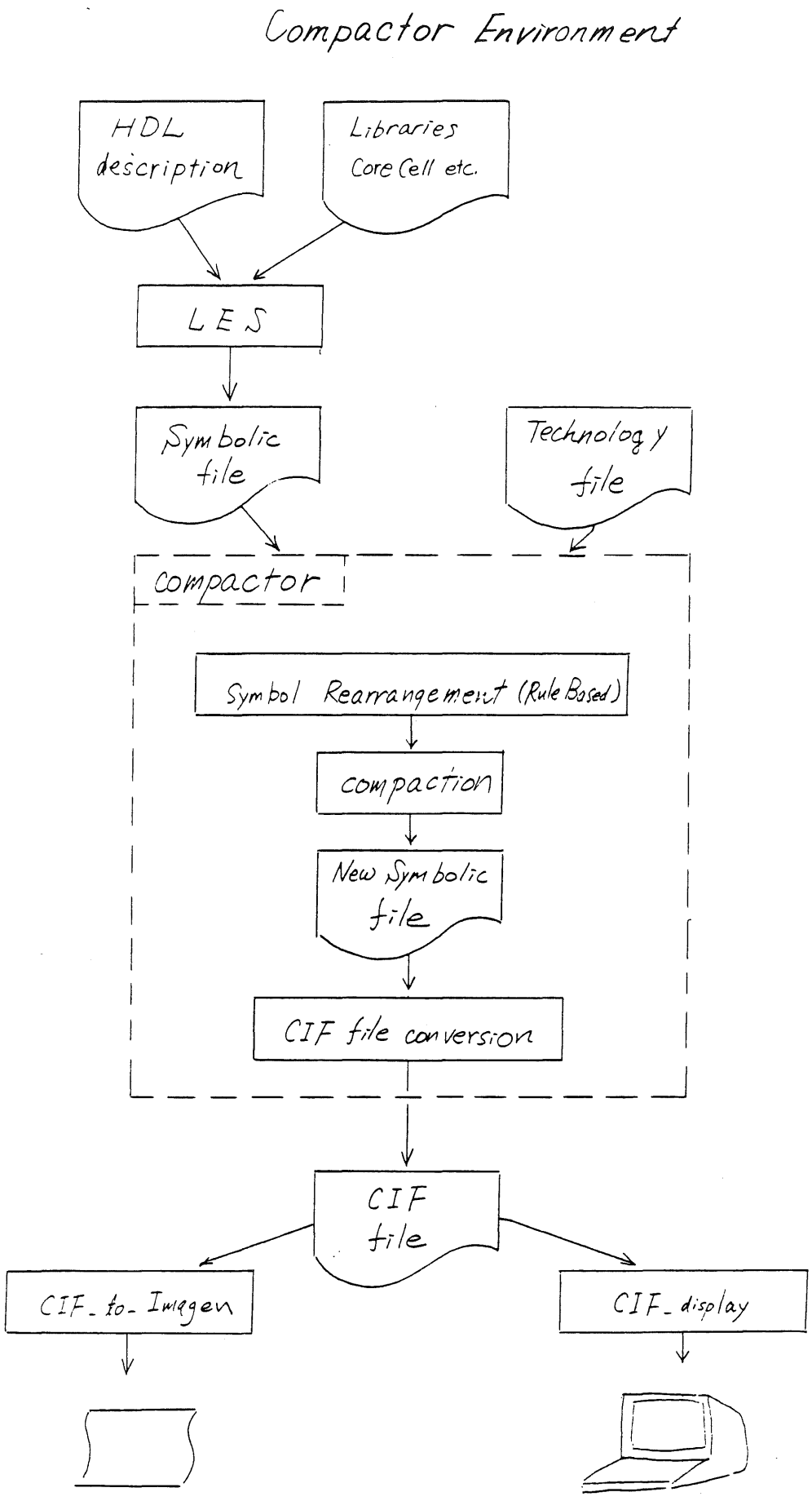
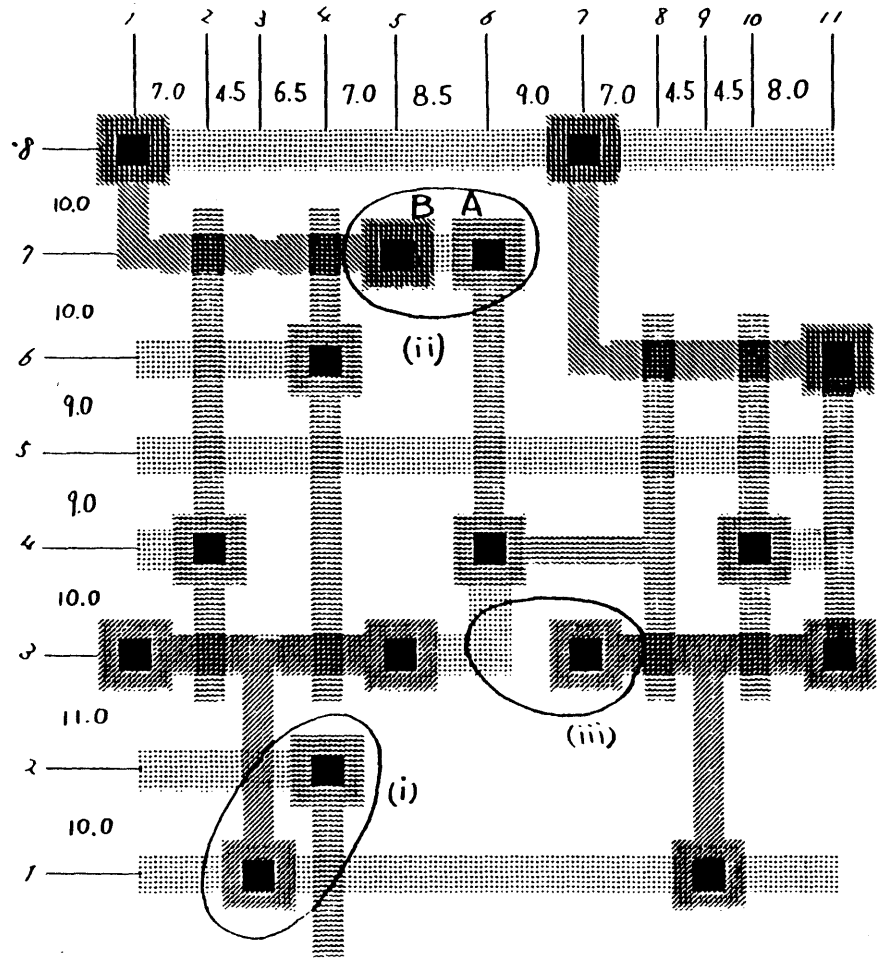
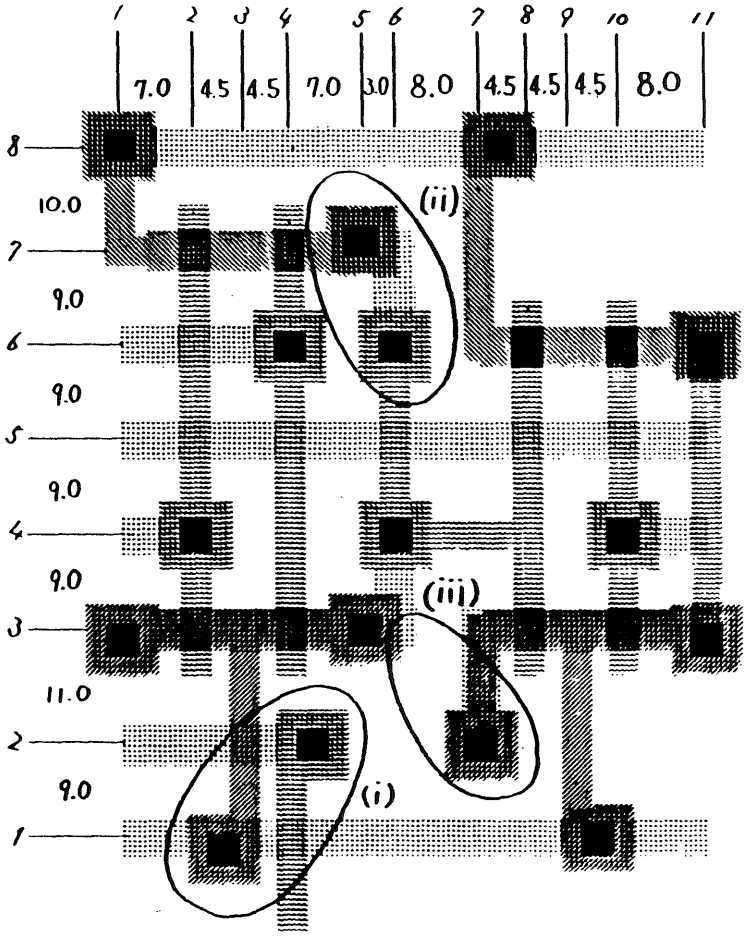


Fig. 1. Two Phase Compactor

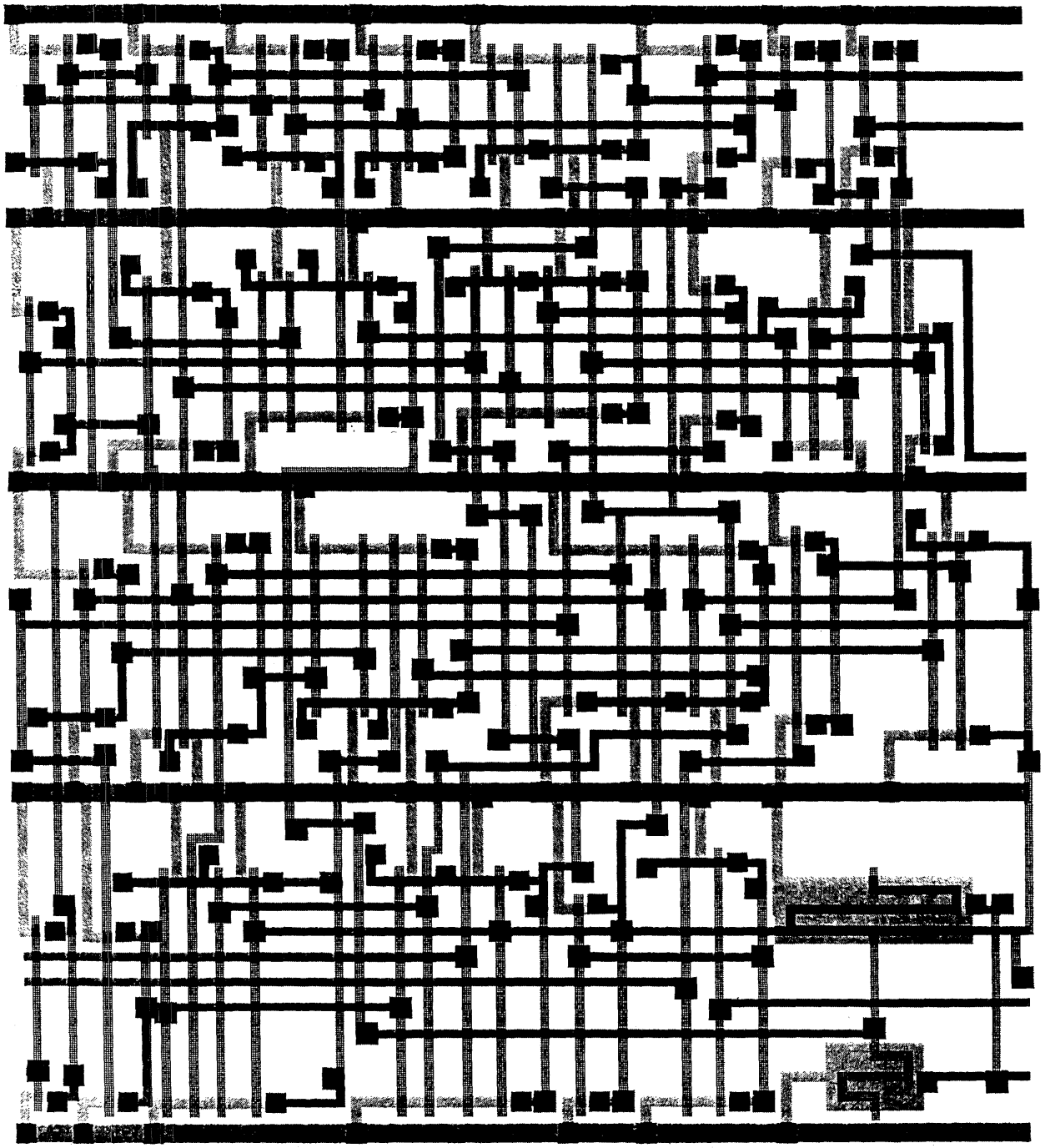


(a) Before



(b) After

Fig. 2 Symbol Rearrangement



Snake Transistor Introduction

$W_p = 32\lambda$

$W_n = 16\lambda$

FIG. 3

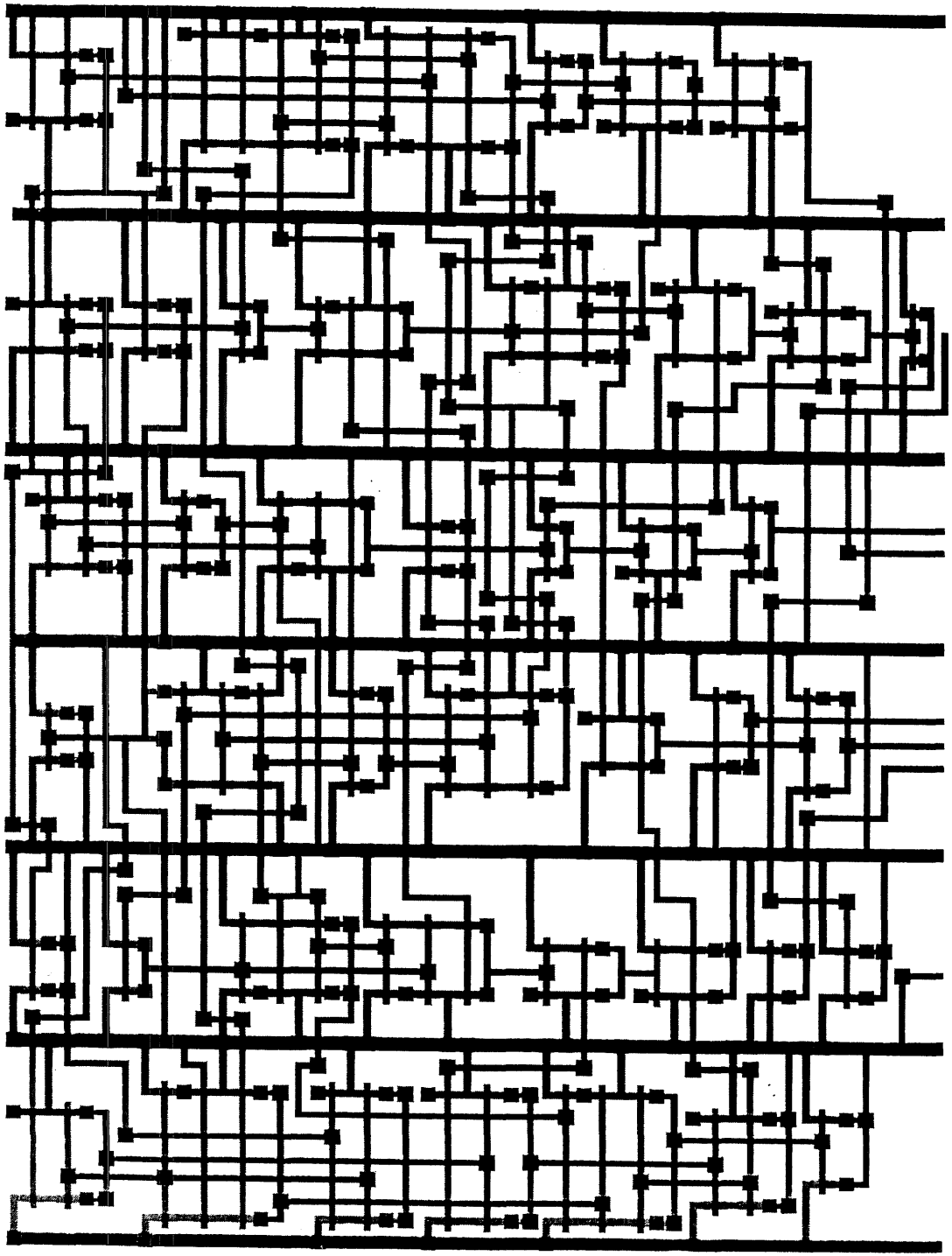


Fig. 4(a) Compaction restricted by virtual grid

Area: $266 \times 352.6 = 93791 \mu m^2$

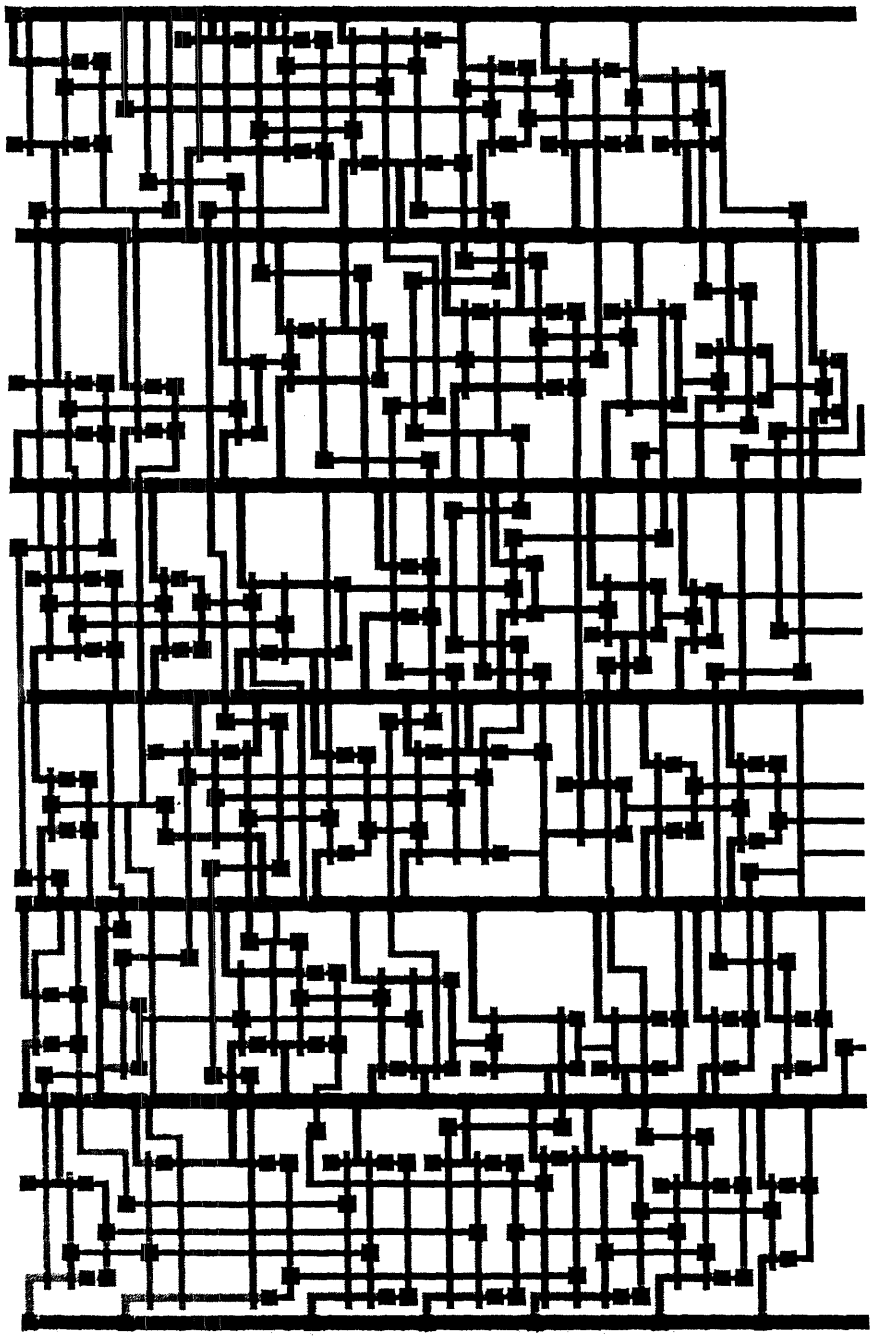


Fig 4(b) Mask layout with symbol rearrangement and group compaction

Area: $206.4 \times 321.2 = 66295.7\ \mu m^2$

new compactor statistics against virtual grid compactor

Circuit	virtual grid compaction	new compactor			
	area w/ symbol rearrangement	w/o symbol rearrangement		w/ symbol rearrangement	
		area	%	area	%
adder4	0.109	0.067	61.5	0.062	56.9
sun1_t	0.090	0.067	74.4	0.064	71.1
big.2	0.234	0.198	84.6	0.190	81.2

Table 1.

area unit : mm^2

in case of 2 micron std proces